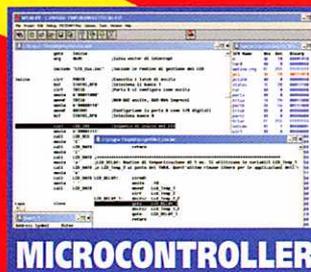
impara

l'elettronica digitale

...e costruisci il tuo LABORATORIO DIGITALE

6,90 €

64

9 771824 363008 41064

Peruzzo & C.

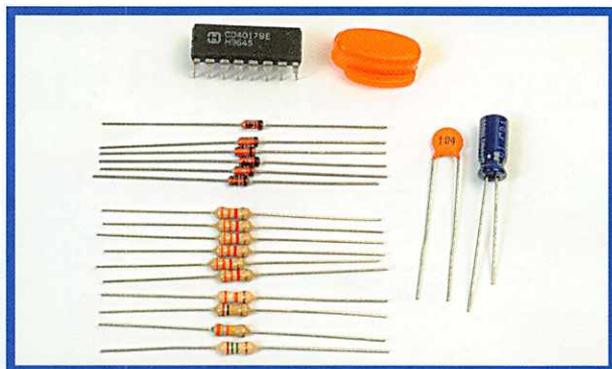
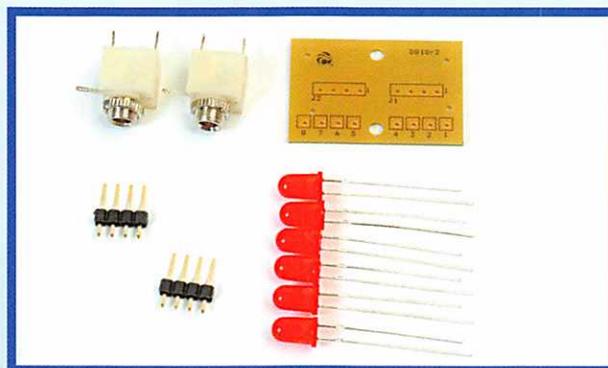
Direttore responsabile:
ALBERTO PERUZZO
Direttore Grandi Opere:
GIORGIO VERCELLINI
Consulenza tecnica e traduzioni:
CONSULCOMP S.n.c.
Pianificazione tecnica
LEONARDO PITTON

Direzione, Redazione, Amministrazione: viale Ercole Marelli 165, Tel. 02/242021, 20099 Sesto San Giovanni (MI). Pubblicazione settimanale. Registrazione del Tribunale di Monza n. 1738 del 26/05/2004. Spedizione in abbonamento postale gr. II/70; autorizzazione delle Poste di Milano n. 163464 del 13/2/1963. Stampa: Grafiche Porpora s.r.l., Cernusco S/N (MI). Distribuzione SO.DI.P. S.p.A., Cinisello Balsamo (MI).

"ELETTRONICA DIGITALE"
si compone di
70 fascicoli settimanali
da suddividere
in 2 raccoglitori.

RICHIESTA DI NUMERI ARRETRATI. Per ulteriori informazioni, telefonare dal lunedì al venerdì ore 9.30-12.30 all'ufficio arretrati tel. 02/242021. Se vi mancano dei fascicoli o dei raccoglitori per completare l'opera, e non li trovate presso il vostro edicolante, potrete riceverli a domicilio rivolgendovi direttamente alla casa editrice. Basterà compilare e spedire un bollettino di conto corrente postale a PERUZZO & C. s.r.l., Ufficio Arretrati, viale Marelli 165, 20099 Sesto San Giovanni (MI). Il nostro numero di c/c postale è 42980201. L'importo da versare sarà pari al prezzo dei fascicoli o dei raccoglitori richiesti, più le spese di spedizione € 3,10 per pacco. Qualora il numero dei fascicoli o dei raccoglitori sia tale da superare il prezzo globale di € 25,82 e non superiore a € 51,65, l'invio avverrà per pacco assicurato e le spese di spedizione ammonteranno a € 6,20. La spesa sarà di € 9,81 da € 51,65 a € 103,29; di € 12,39 da € 103,29 a € 154,94; di € 14,98 da € 154,94 a € 206,58; di € 16,53 da € 206,58 in su. Attenzione: ai fascicoli arretrati, trascorse dodici settimane dalla loro distribuzione in edicola, viene applicato un sovrapprezzo di € 0,52, che andrà pertanto aggiunto all'importo da pagare. Non vengono effettuate spedizioni contrassegno. Gli arretrati di fascicoli e raccoglitori saranno disponibili per un anno dal completamento dell'opera. IMPORTANTE: è assolutamente necessario specificare sul bollettino di c/c postale, nello spazio riservato alla causale del versamento, il titolo dell'opera nonché il numero dei fascicoli e dei raccoglitori che volete ricevere.

IN REGALO in questo fascicolo

1 Condensatore da 10 µF elettrolitico
1 Circuito integrato 4017
1 Condensatore da 100 nF ceramico
6 Diodi 1N4148
1 Resistenza da 39 K 5% 1/4 W
6 Resistenze da 3K3 5% 1/4 W
1 Resistenza da 330 K 5% 1/4 W
1 Resistenza da 100 K 5% 1/4 W
1 Resistenza da 1M K 5% 1/4 W
1 Tasto di plastica

IN REGALO nel prossimo fascicolo

1 Scheda DG10
2 Jack femmina da pannello
2 Connettori da c.s. maschio, diritti a 4 vie
6 LED rossi

COME RACCOGLIERE E SUDDIVIDERE L'OPERA NELLE 4 SEZIONI

L'Opera è composta da 4 sezioni identificabili dalle fasce colorate, come indicato sotto. Le schede di ciascun fascicolo andranno suddivise nelle sezioni indicate e raccolte nell'apposito raccoglitore, che troverai presto in edicola. Per il momento, ti consigliamo di suddividere le sezioni in altrettante cartellette, in attesa di poterle collocare nel raccoglitore. A prima vista, alcuni numeri di pagina ti potranno sembrare ripetuti o sbagliati. Non è così: ciascuno fa parte di sezioni differenti e rispecchia l'ordine secondo cui raccogliere le schede.
Per eventuali domande di tipo tecnico scrivere al seguente indirizzo e-mail: elettronicadigitale@microrobots.it

Hardware Montaggio e prove del laboratorio

Digitale di base Esercizi con i circuiti digitali

Digitale avanzato Esercizi con i circuiti sequenziali

Microcontroller Esercizi con i microcontroller

Generatore al quarzo (II)

La scheda DG17 contiene molti componenti che verranno forniti in parte con questo fascicolo e in parte con il fascicolo successivo.

Componenti allegati a questo fascicolo.

Integrato 4017.

I componenti

In questo caso i componenti sono numerosi e di seguito ne riportiamo l'elenco: un condensatore elettrolitico da 10 μF, un circuito integrato 4017, un condensatore da 100 nF, sei diodi del tipo 1N4148, una resistenza da 39 K, sei resistenze da 3K3, una resistenza da 330 K, una resistenza da 100 K, una resistenza da 1 M e un tasto di plastica per attivare il pulsante. Benché il montaggio di questa scheda risulti laborioso, non è difficile, ma deve essere eseguito con attenzione, in modo che ogni componente sia ben collocato nel posto a lui assegnato e con la polarità corretta.

Schema elettrico.

Resistenze da 3K3.

Dettaglio della resistenza R3.

Resistenze

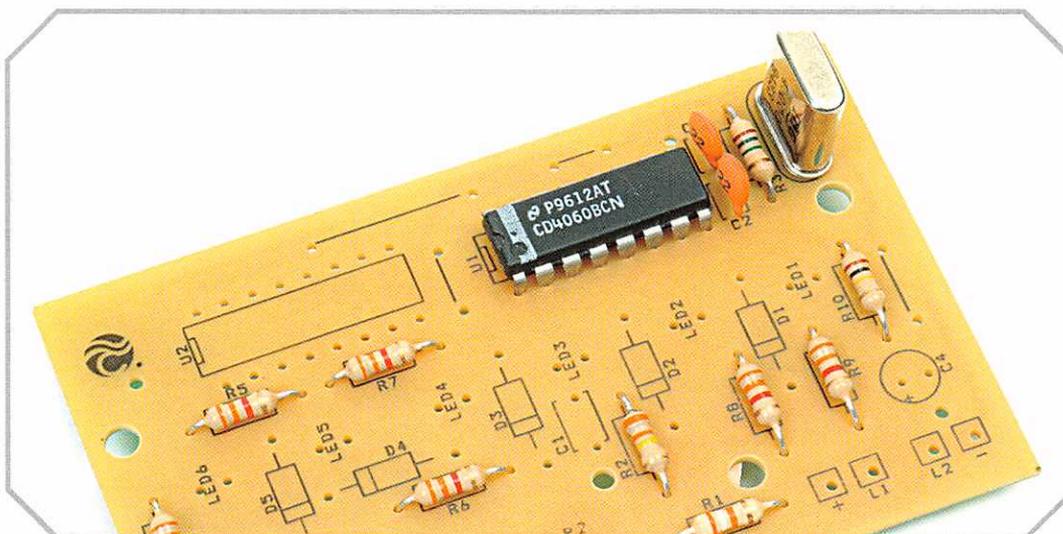
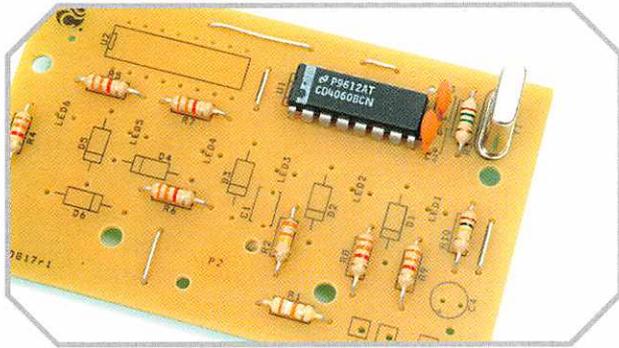
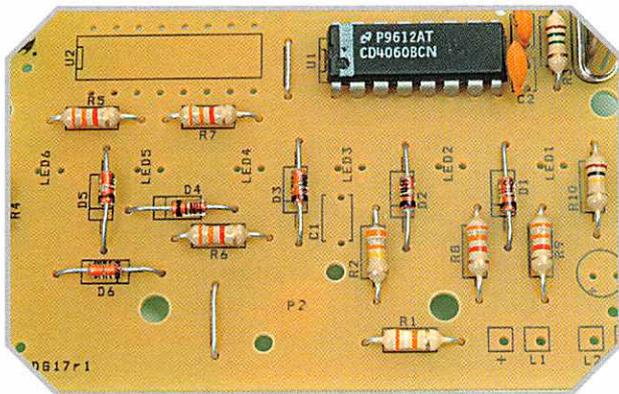
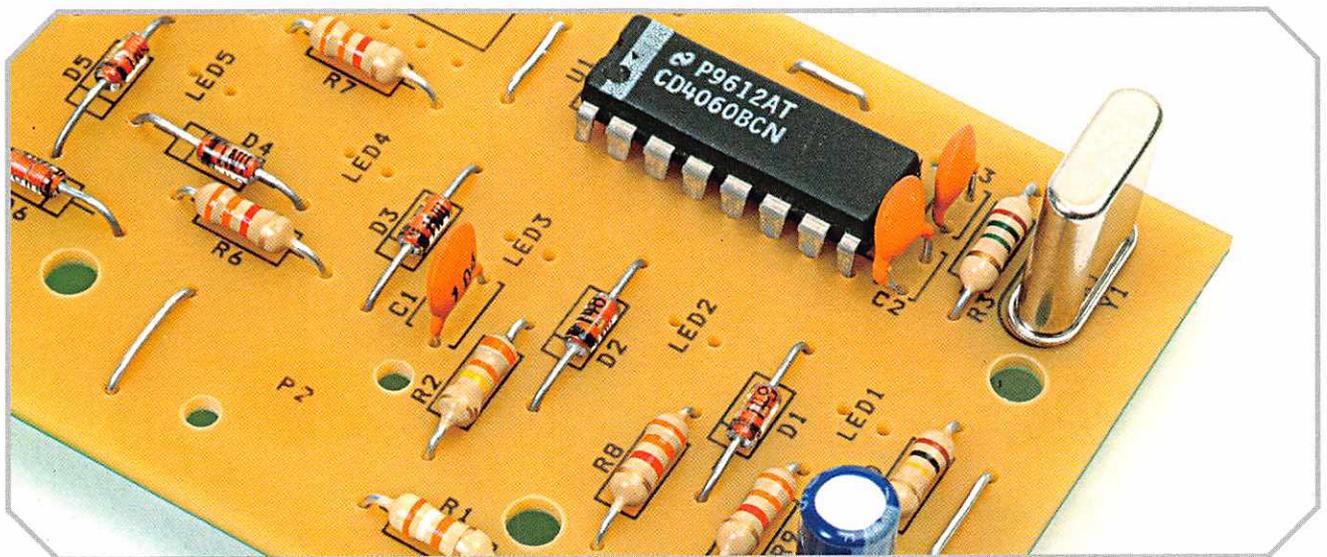
Su questa scheda è necessario inserire e saldare dieci resistenze con cinque valori diversi, quindi dovremo essere ordinati per evitare errori. Le prime resistenze da montare sono quelle che corrispondono ai riferimenti da R4 a R9, tutte da 3K3 (arancio, arancio, rosso), dopo averne saldati i terminali procederemo al taglio della parte eccedente dei terminali stessi. Monteremo poi la resistenza da 1 M (marrone, nero, verde) tra il cristallo di quarzo e i condensatori C2 e C3. Le resistenze successive, R1 da 39 K (arancio, bianco, arancio), R2 da 330 K (arancio, arancio, giallo) e R10 da 100 K (marrone, nero, giallo) verranno inserite nelle rispettive posizioni.

La parte eccedente dei terminali deve essere conservata perché si potrà utilizzare come indicheremo di seguito.

Ponticelli

L'elevato numero di componenti necessari per il funzionamento del circuito, obbliga a eseguire cinque ponticelli su questa scheda, i quali si possono realizzare con le rimanenze dei reofori che derivano dal taglio dei terminali delle resistenze.

Diodi

Questo circuito ha sei diodi di segnale del tipo 1N4148. Questi componenti hanno polarità, il

Scheda DG17 con tutte le resistenze montate.

Questa scheda contiene cinque ponticelli.

Diodi 1N4148.

loro catodo è identificato da una banda vicina a questo terminale, che nella serigrafia si rappresenta con un tratto che attraversa il disegno del componente e anch'esso è il più vicino al terminale del catodo. Alcuni costruttori di questi diodi utilizzano due bande gialle per siglare i componenti, in questo caso quella che corrisponde al catodo è la banda più larga. I sei diodi si inseriscono rispettandone la polarità nei fori indicati dalle serigrafie da D1 a D6.

Condensatori

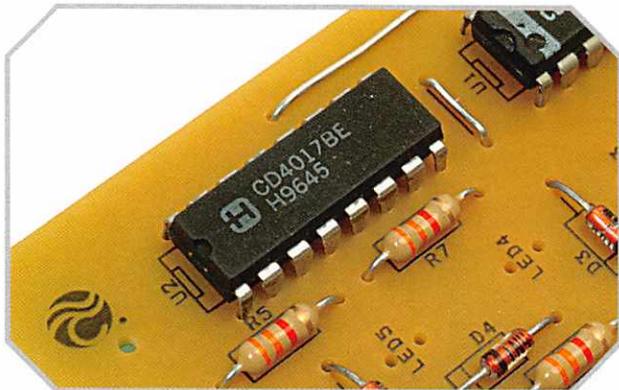
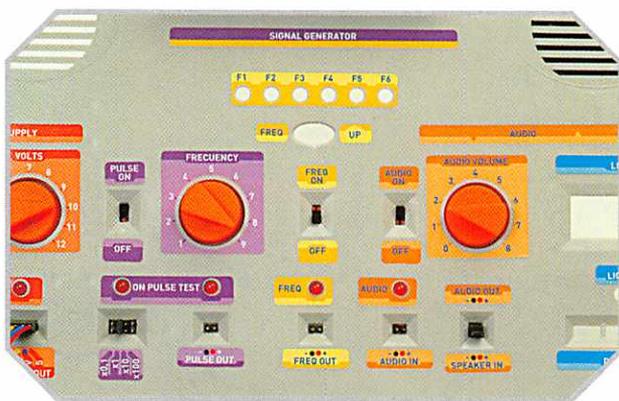
Il condensatore C1 non ha polarità, quindi è sufficiente inserirne i terminali, invece C4 è del tipo elettrolitico e bisogna tener conto della sua polarità: il terminale positivo è indicato sulla scheda col segno +.

L'integrato 4017

Questo integrato ha assegnato come sigla U2 e si monta come d'abitudine, identificando con attenzione il terminale 1 per poterlo orientare correttamente e allineando bene i piedini in modo che tutti entrino senza difficoltà nei fori del circuito stampato.

Il circuito

Dopo aver spiegato le funzioni dei due circuiti integrati utilizzati su questa scheda, dobbiamo

Condensatori C1 e C4.

Integrato 4017 sulla scheda.

In questa zona verrà montata la scheda DG17.

parlare del funzionamento del sistema di selezione dell'uscita, che utilizza dodici diodi: sei LED e sei diodi di segnale.

Per iniziare la spiegazione concentriamoci su una delle uscite. Immaginiamo che sia attiva l'uscita Q0 del 4017, ricordando che può essere attiva solamente una delle uscite, tramite la resistenza R9 Q0 polarizza direttamente il LED D1 e il segnale raggiunge l'uscita della DG17 terminale L1; se il LED 1 non fosse installato il livello su questa uscita sarebbe sempre alto (non consideriamo per il momento la caduta di tensione sul diodo).

Vediamo ora l'uscita del 4060, in questo caso la Q14 del 4060, questa uscita sta fornendo un segnale a onda quadra da 244 Hz e quando è a livello alto il LED 1 non è polarizzato, dato che all'altro capo ha un livello alto, l'uscita su L1 quindi ha livello alto. Quando invece il livello del segnale è basso, il LED 1 è polarizzato e conduce (quindi si illumina), e il livello sull'ingresso L1 diventa basso (non consideriamo per il momento la caduta di tensione sul diodo), a causa della resistenza R10 e del resto del circuito a cui è collegato. Quindi, tralasciando sempre le cadute di tensione sui diodi, il segnale di uscita Q14 del 4060 si trasmette all'uscita L1.

È necessario tener presente che le uscite del 4060 sono tutte attive e forniscono segnale. La selezione si fa con le uscite del 4017 delle quali solamente una è attiva; osservate nello schema che l'uscita Q6 del 4017 è collegata al terminale di reset dello stesso, in questo modo il contatore attiva solamente le uscite utilizzate, passando dalla Q5 alla Q0.

Vista generale del laboratorio con la scheda a fianco.

Temporizzatore doppio

Questo circuito si utilizza per emettere un suono durante un determinato periodo, trascorso il quale il suono cambia. Una delle applicazioni potrebbe essere avvisare i non vedenti, ad esempio, che il semaforo permette il passaggio, mentre il secondo suono informa che il tempo sta passando e il semaforo sta per bloccare il passaggio.

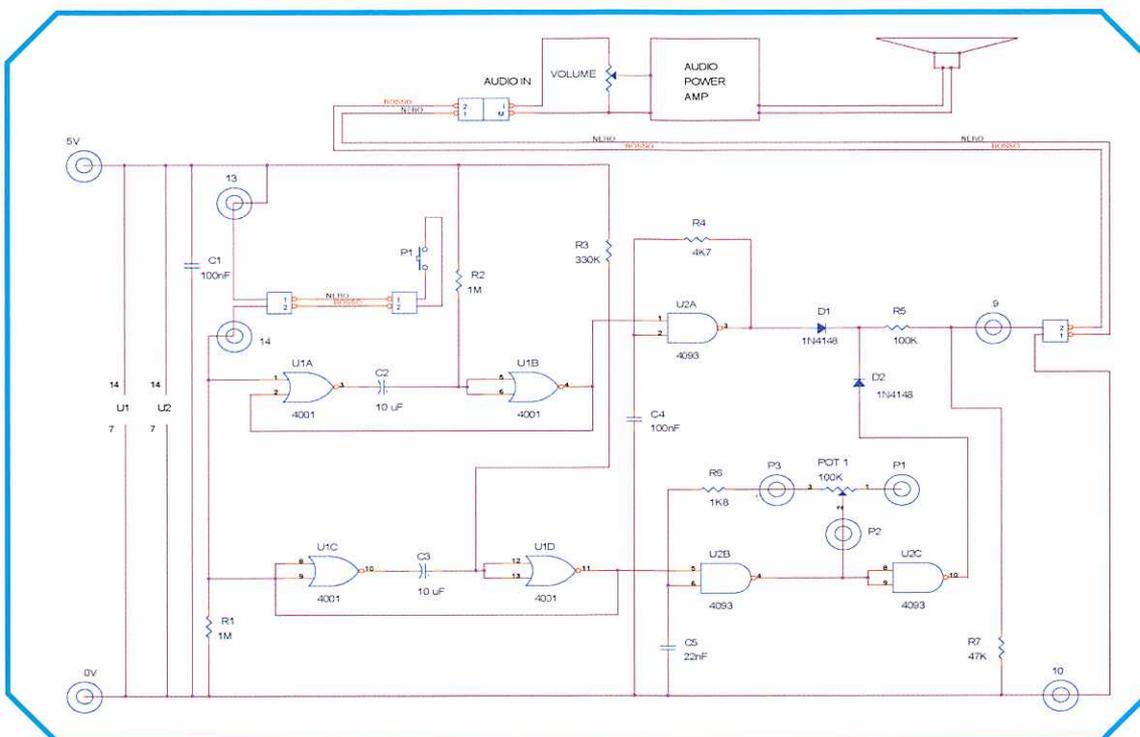
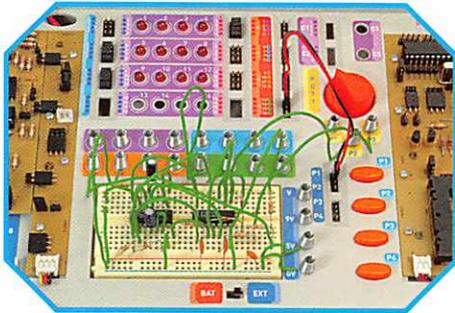
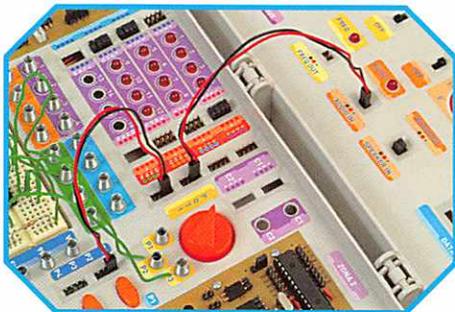
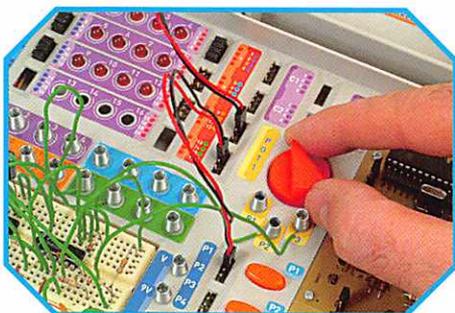
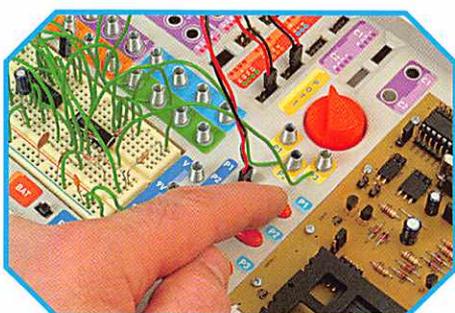
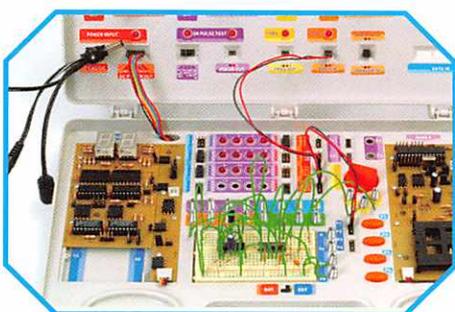
Il circuito

Anche se il circuito completo è nuovo lo possiamo scomporre in parti che sono già note.

Le porte U1A e U1B, unite al condensatore C2 e alla resistenza R2, formano un temporizzatore monostabile. L'uscita, che in stato di riposo è a livello basso, si attiva passando a livello alto; quando il suo terminale 1 si porta per un attimo a livello alto, questo livello alto si ottiene azionando il pulsante P1. L'uscita di questo monostabile si utilizza per attivare l'oscillatore astabile formato attorno alla porta NAND U2A, la cui uscita viene portata, tramite un diodo, all'ingresso dell'amplificatore audio, AUDIO IN. Le resistenze R5 e R7 formano un partitore di tensione che attenua il segnale prima che sia applicato all'amplificatore audio.

Osservando attentamente il circuito vedremo che le porte U1C e U1D formano un temporizzatore monostabile la cui uscita, quando è a livello alto, attiva un altro oscillatore, che in questo caso dispone di un potenziometro

Componenti sulla scheda Bread Board.

Cablaggio interno della scheda.

Schema elettrico.

Collegamento del pulsante.

Collegamento all'amplificatore audio.

Regolazione della frequenza.

Attivazione del circuito.

Esperimento completato.

che ci permette di modificarne la frequenza per ottenere un suono diverso dal precedente. Questo segnale viene miscelato col precedente tramite il diodo D2, prima di essere applicato all'ingresso dell'amplificatore audio.

Montaggio

Il montaggio si esegue come d'abitudine, evitando di scambiare i due circuiti integrati tra loro e facendo particolare attenzione alla polarità dei condensatori elettrolitici e dei diodi. Il cavo di collegamento all'ingresso dell'amplificatore audio deve essere collegato facendo attenzione ai colori, a questo scopo facciamo riferimento alle fotografie. Per il collegamento del pulsante, invece, non è necessario fare attenzione alla polarità. L'alimentazione dei due integrati è uguale, il terminale 14 è il positivo e il 7 è il negativo; il circuito deve essere alimentato a 5 V.

Funzionamento

Dopo aver montato il circuito potremo verificarne il funzionamento. È necessario collegare l'alimentazione e azionare l'amplificatore con AUDIO ON, posizionando il comando del volume vicino al minimo. Azionando il pulsante si attivano i due monostabili contemporaneamente, la porta U1B rimane a livello alto per un tempo più lungo della U1D, dato che la resistenza R2 è maggiore della R3, mentre i condensatori sono uguali. Questi ultimi componenti sono quelli che dovranno essere modificati per eventuali applicazioni specifiche. Dopo aver premuto il pulsante i due generatori di suono devono funzionare contemporaneamente, ma uno di essi deve funzionare prima dell'altro.

LISTA DEI COMPONENTI

U1	Circuito integrato 4001
U2	Circuito integrato 4093
R1, R2	Resistenza 1 M (marrone, nero, verde)
R3	Resistenza 330 K (arancio, arancio, giallo)
R4	Resistenza 4K7 (giallo, viola, rosso)
R5	Resistenza 100 K (marrone, nero, giallo)
R6	Resistenza 1K8 (marrone, grigio, rosso)
R7	Resistenza 47 K (giallo, viola, arancio)
C1 , C4	Condensatore 100 nF
C2, C3	Condensatore 10 µF, elettrolitico
C5	Condensatore 22 µF
D1, D2	Diodo 1N4148

PicBasic Plus Lite

L'assembler è un linguaggio che abbiamo studiato per programmare i PIC, ma essendo un linguaggio di basso livello, utilizzarlo per risolvere applicazioni che esigono molta programmazione potrebbe risultare complicato.

PICBASIC PLUS LITE

Oggi è possibile trovare dei software per programmare i PIC con linguaggio di alto livello. Esistono dei compilatori e degli assemblatori che convertono in linguaggi come il C o il Basic in codice macchina, per poter scrivere i programmi sul PIC.

Con il PICBASIC PLUS LITE possiamo programmare in linguaggio Basic utilizzando i dispositivi del PIC16F87X, l'unico inconveniente o limitazione è che i programmi devono essere molto brevi, non devono superare le 20 linee.

Il PicBasic Plus Lite.

Installazione

Il PICBASIC PLUS LITE si trova sul primo CD fornito insieme all'opera all'interno della cartella di Programmi, compresso come un file ZIP di nome "lite_plus". Per decomprimerlo cliccate due volte col tasto sinistro del mouse sul file e vi apparirà una finestra come quella della figura in basso. Dovremo inserire l'indirizzo dove vogliamo che il file venga decompresso e cliccare su Unzip. Quando il processo sarà terminato cliccate su Close per chiudere la finestra. All'indirizzo selezionato verrà creata una directory PBPLUS_LITE in cui si trova il file eseguibile PBP_Lite.

Mandando in esecuzione questo file si entra direttamente nel programma e ci verrà presentata una finestra informativa dello stesso, con la versione e le caratteristiche del compilatore. Per iniziare a utilizzare il programma clicchiamo sul pulsante OK della finestra.

Dobbiamo decomprimere il file per poter installare il programma.

Facciamo conoscenza con il programma

Per familiarizzare col programma vedremo a che cosa servono ciascuno dei menù a cui possiamo accedere tramite il menù principale.

Videata iniziale del PicBasic Plus Lite.

Il menù di File (File)

In questo menù troviamo le operazioni tipiche che si possono eseguire sui file. Possiamo creare un nuovo file (New), o aprirne uno già esistente (Open), salvare i cambiamenti eseguiti sul file (Save), salvare le variazione come un file nuovo (Save As...) e stampare il file (Print). Per quanto riguarda le funzioni non direttamente riferite ai file possiamo configurare la stampante (Print Setup...), configurare l'editor (Editor Option) e uscire dal programma (Exit).

Nella configurazione dell'editor possiamo selezionare diverse opzioni per il lavoro con quest'ultimo (margini, numeri di linee, di pagine, ecc.) e anche scegliere il formato delle linee di codice.

Il menù di Edizione (Edit)

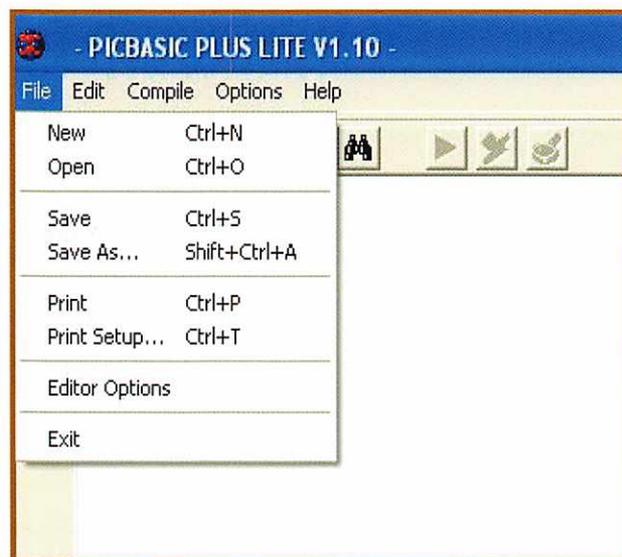
Anche questo menù è tra quelli che normalmente si trovano all'interno di qualsiasi programma. In esso troviamo le funzioni di Edit come quella di ripetere gli ultimi cambiamenti (Redo) o annullare le ultime modifiche (Undo), tagliare (Cut), copiare (Copy), incollare (Paste) o selezionare tutto (Select All), trovare un testo (Find...) o sostituirlo (Replace...) e infine, una funzione per passare da minuscolo a maiuscolo (To UpperCase) e viceversa da maiuscolo a minuscolo (To LowerCase).

Il menù di compilazione (Compile)

Questo menù è specifico dei programmi di programmazione. In esso si contemplano le opzioni di compilazione del programma. Mediante Compile Basic si esegue la compilazione del codice che è stato confezionato nell'editor, ovvero si tradurrà il programma scritto in Basic in linguaggio assembler. Da questo processo, quindi, otterremo il file in assembler ".asm", il file in esadecimale da trasferire al microcontroller ".hex", il file con il listato del codice ".cod" e quello che riporta gli errori rilevati nella compilazione. Viene anche creato un file proprio del programma con estensione ".pbp". Tutti questi file vengono generati nella directory Samples all'interno della cartella del programma PBPLUS_LITE.

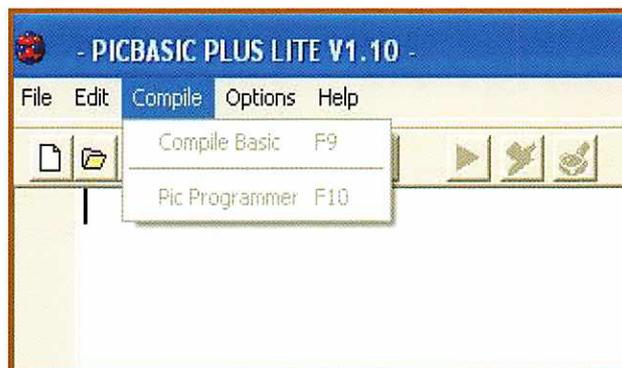
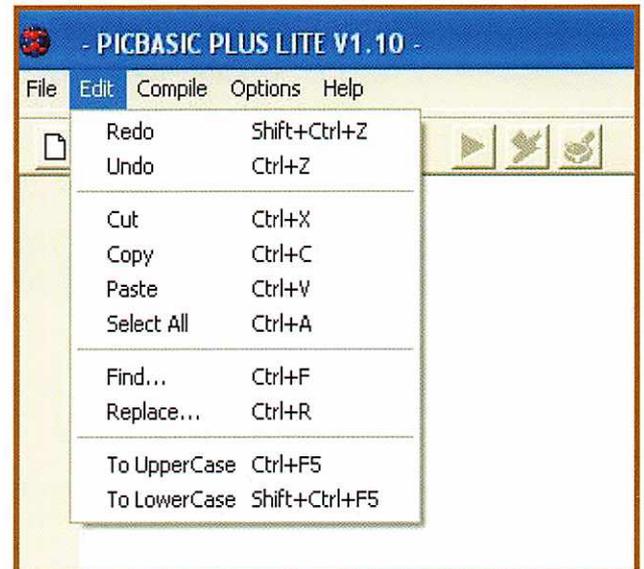
Mediante l'opzione Pic Programmer potremo scrivere sul microcontroller dopo aver compilato il programma. Il programmatore lo dobbiamo scegliere nel menù che spiegheremo di seguito.

Il menù delle Opzioni (Options)

La prima opzione di questo menù Programmer Options serve per scegliere il programma-

Funzioni del menù File.

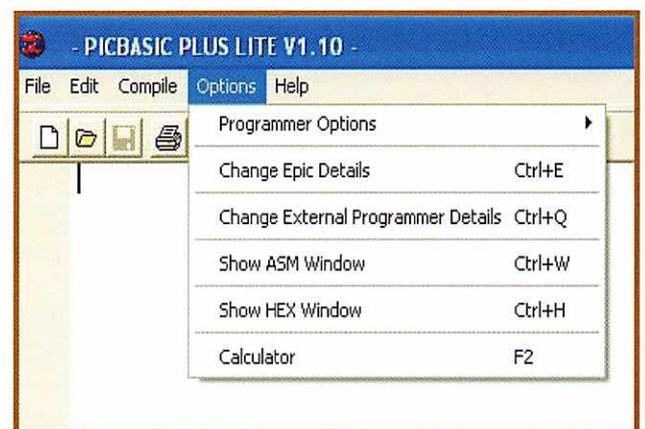
tore o l'elemento hardware che permetta di scrivere il nostro programma, una volta compilato, sul microcontroller. È possibile eseguire una scelta fra tre opzioni, anche se nel nostro caso è indifferente qualsiasi sia la scelta, dato che utilizzeremo il software IC_Prog come software esclusivo di scrittura. Le due opzioni successive Change Epic Details e Change External Programmer Details servono per selezionare l'indirizzo dove risiedono i programmi che fanno riferimento a questi programmatori, ma neanche questi rivestono molta importanza. Le opzioni successive, invece, ci possono essere molto utili. Show ASM Window e Show HEX Window mostrano entrambe una finestra con il codice in assembler (asm) generato e con il codice in linguaggio macchina (hex). In questo modo potremo vedere l'equivalenza tra i tre linguaggi, benché per il linguaggio macchina sia piuttosto difficile capire qualcosa del programma, in quanto tutte le istruzioni sono state trasformate da mnemonico (sia esso del Basic o quello dell'assembler) in una sequenza di caratteri alfanumerici. L'ultima opzione di questo menù è Calcolatore, che è un accesso diretto alla calcolatrice interna di Windows. È utile avere una calcolatrice a portata di mano per eseguire i calcoli delle temporizzazioni, delle conversioni tra formato binario ed esadecimale, o qualsiasi altro calcolo di una certa complessità.

Il menù di Aiuto

Questo menù presenta solamente due opzioni, Syntax e About. La prima opzione apre il manuale del programma LetPicBasicPlus su Internet Explorer. Ha un formato ".htm" ed è un manuale completo orientato principalmente alla programmazione in Basic interrelazionato con le funzioni del microcontroller. La seconda opzione apre la videata iniziale del programma dove appare la versione del compilatore.

Funzioni del menù Edit.

La barra degli strumenti

Le icone della barra degli strumenti riassumono le funzioni più importanti tra quelle che abbiamo appena visto. Posizionando il cursore del mouse su una di queste icone viene visualizzato un testo che ne descrive la funzionalità.

Il menù di Compilazione.

Il menù di Opzioni.

Barra degli strumenti.

Il menù di Aiuto.

Conclusioni

Il programma PICBASIC PLUS LITE è un esempio di come programmare un PIC con un linguaggio di programmazione di alto livello. Esistono compilatori per altri linguaggi, ma il Basic è il più diffuso dopo l'assembler stesso. Presenteremo alcuni esercizi che servono da esempio per il funzionamento del programma, comunque dovrete essere voi stessi ad approfondire il linguaggio di programmazione. Nella cartella Programmi del primo CD allegato all'opera, oltre al programma LetPicBasic Plus troverete anche una guida o manuale chiamato "picbasic plus lite manual" che vi servirà come aiuto per imparare a programmare in linguaggio Basic sulla piattaforma presentata.

Questo software è un altro utile e pratico strumento e ne verrà continuata la presentazione nei prossimi fascicoli.

PICBASIC PLUS LITE COMPILER

Please Note.

The LITE version of the PICBASIC PLUS compiler offers an opportunity to write a program for the PIC first hand. See for yourself how powerful and easy to use the compiler really is.

The LITE version is limited to 20 lines of code, with only one commnand per line, as opposed to multiple commands, and unlimited lines in the full version. Anmother limitation is the type of PIC supported. The LITE version is compatable with the ever-popular 16F84 and 16F877 devices, with crystal frequencies of 4MHz and 20MHz only. The full version is compatable with all the 14-bit core devices, and 6 crystal frequencies. Another limitation is the lack of any graphic LCD commands. Other than that, the compiler is fully operational.

Although every precaution has been taken with the preparation of this manual to ensure that any projects, designs or programs enclosed, operate in a correct and safe manner. The author and publisher assume no responsibility for errors or omissions. Neither is any liability assumed for the failure of any project, design or program, or any damage caused to equipment that it may be connected to, or used in combination with.

Copyright Crownhill Associates. All right reserved. No part of this publication may be reproduced, stored in a retrieval system, or distributed in any form or by any means without the written permission of the publisher or author.

The Microchip logo and name are registered trademarks of Microchip Technologies Inc.

The EPIC^tm programmer is a trade name of microEngineering Labs inc.

Manuale di aiuto del PicBasic Plus Lite.

Primo programma con display LCD

Conosciamo già il funzionamento e l'architettura interna del nostro display LCD, quindi ci rimane solamente da sapere come lavorare con esso. Realizzeremo un semplice progetto con cui interagiremo con il modulo per abituarci poco a poco alla sua gestione.

Enunciato

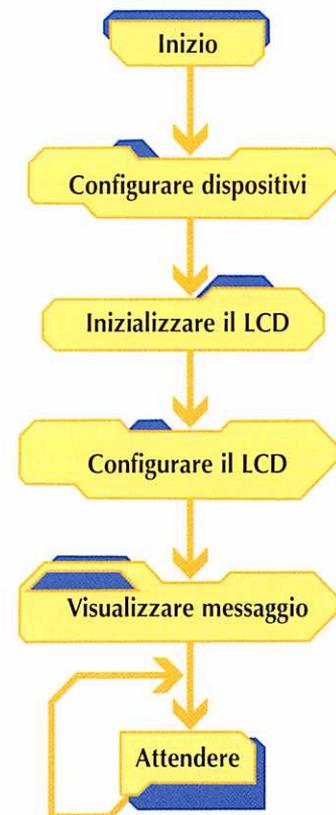
Si vuole visualizzare sul modulo LCD un messaggio, ad esempio: "Ciao". Dobbiamo applicare tutte le conoscenze acquisite per risolvere questo semplice programma e provarlo sul laboratorio.

Nella libreria "lcd_cxx.in", che potete trovare nella cartella LCD nel secondo CD allegato all'opera, esiste una routine che invia impulsi all'ingresso di abilitazione.

Organigramma

Osservando l'organigramma che abbiamo predisposto per risolvere l'applicazione, verificherete che l'unica difficoltà del programma è quella di interagire per la prima volta con il nostro modulo LCD. Provate a pianificare una soluzione al progetto e confrontatela con quella che svilupperemo di seguito.

Codice

Apriremo l'editor di testo e mediante dei commenti spiegheremo ciò che dovrà contenere il file e le sue funzionalità. Intesteremo il file definendo il PIC e le librerie con le quali vogliamo lavorare. Solitamente lavoriamo so-

Organigramma dell'applicazione.

ciao.asm - Blocco note

File Modifica Formato Visualizza ?

```
;        ESERCIZIO: CIAO
;       -----------------
;
;Questo esempio ci vuole introdurre alla gestione del display LCD, per la visualizzazione
;dei diversi messaggi (ad esempio Ciao).

                    List     p=16F870          ;Tipo di processore
                    include  "P16F870.INC"     ;Definizione dei registri interni

Lcd_var             equ      0x20              ;Variabili (2) delle routine di gestione del LCD

                    org      0x00              ;Vector di Reset
                    goto     Inizio
                    org      0x05              ;Salva vector di interrupt

                    include  "LCD_Cxx.inc"     ;Include le routine di gestione del LCD
```

Intestazione del programma.

lcd_cxx.asm - Blocco note

File Modifica Formato Visualizza ?

```
;                         LCD_CXX.INC
;
;
;
;L'insieme di routine che presentiamo di seguito, permette di realizzare le operazioni fondamentali
;per il controllo del modulo di visualizzazione LCD. Questo file si deve includere nei futuri
;programmi sorgente mediante la direttiva INCLUDE.

#define ENABLE          bsf PORTA,2        ;Attiva segnale E
#define DISABLE         bcf PORTA,2        ;Disattiva segnale E
#define LEGGERE         bsf PORTA,1        ;Poni LCD in Modo RD
#define SCRIVERE        bcf PORTA,1        ;Poni LCD in Modo WR
#define OFF_COMANDO     bcf PORTA,0        ;Disattiva RS (modo comando)
#define ON_COMANDO      bsf PORTA,0        ;Attiva RS (modo dato)

            CBLOCK  Lcd_var              ;Inizio delle variabili. Sarà il primo
            Lcd_Temp_1                   ;indirizzo libero disponibile
            Lcd_Temp_2
            ENDC

;************************************************************************
```

Definizione del blocco delle variabili nella libreria.

lamente con quella che contiene le definizioni dei registri del PIC "P16F870.inc", ma in questo caso abbiamo bisogno anche della libreria che contiene anche tutte le subroutine di gestione del display LCD "lcd-cxx.inc".

Continueremo l'intestazione con le direttive di intestazione del programma nella memoria ORG e per concludere questa prima parte del programma, definiremo la variabile "Lcd_var", assegnandola a una posizione di memoria. È necessario definire questa variabile ogni volta che lavoriamo con il modulo LCD, in quanto nella libreria di gestione del display è contemplato il suo utilizzo. In questa libreria viene definita una struttura, cioè un blocco di variabili raggruppate sotto lo stesso nome: "Lcd_var". Questo blocco contiene due variabili, "Lcd_Temp_1" e "Lcd_Temp_2", quindi in realtà nel programma che svilupperemo quando definiamo "Lcd_var" stiamo riservando posizioni di memoria per le due variabili prima descritte. È importante mantenere un certo ordine al momento di confezionare un codice. Quindi, tutte le chiamate alle librerie, devono essere raggruppate, così come tutte le definizioni di variabili, commenti, ecc. Questo ordine migliorerà la struttura del programma rendendone più facile il successivo studio e la comprensione.

Configurazione dei dispositivi

Configureremo i dispositivi del microcontroller che vogliamo utilizzare. In questo caso lavoreremo solamente con le porte di ingresso/uscita. Configureremo la porta B come uscita e la porta A come mista, cioè RA0-RA2 come uscite e RA3-RA4 come ingressi digitali.

Visualizzazione del messaggio

Per risolvere l'applicazione dobbiamo visualizzare il messaggio "Ciao" sul modulo LCD. A questo scopo lo dobbiamo inizializzare dato che la routine che lo inizializza si trova definita nella libreria "lcd_cxx.inc", eseguiremo una

ciao.asm - Blocco note

File Modifica Formato Visualizza ?

```
Inizio          clrf    PORTB           ;Cancella i latch di uscita
                bsf     STATUS,RP0      ;Seleziona il banco 1
                clrf    TRISB           ;Porta B si configura come uscita
                movlw   b'00011000'
                movwf   TRISA           ;RA0-RA2 uscite, RA3-RA4 ingressi
                movlw   b'00000110'
                movwf   ADCON1          ;Configuriamo la porta A come I/O digitali
                bcf     STATUS,RP0      ;Seleziona banco 0
```

Configurazione dei dispositivi.

ciao.asm - Blocco note

File Modifica Formato Visualizza ?

```
                call    LCD_INI         ;Sequenza di inizio del LCD
                movlw   b'00001111'
                call    LCD_REG         ;Invia istruzione: LCD ON, cursore ON e blink ON
                movlw   'C'
                call    LCD_DATO        ;Visualizza C
                movlw   'i'
                call    LCD_DATO        ;Visualizza i
                movlw   'a'
                call    LCD_DATO        ;Visualizza a
                movlw   'o'
                call    LCD_DATO        ;Visualizza o
                movlw   ' '
                call    LCD_DATO        ;Visualizza spazio vuoto

Loop            sleep                   ;Imposta in Standby
                goto    Loop            ;Ritorna in standby
```

Visualizzazione del messaggio.

Build Results

```
Building CIAO.HEX...

Compiling CIAO.ASM:
Command line: "C:\PROGRA~1\MPLAB\MPASMWIN.EXE /p16F870 /q C:\PROGRA~1\MPLAB\PROGETTI\CIAO.ASM"
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\CIAO.ASM 20 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\CIAO.ASM 22 : Register in operand not in bank 0.  Ensure
Message[302] C:\PROGRA~1\MPLAB\PROGETTI\CIAO.ASM 24 : Register in operand not in bank 0.  Ensure

Build completed successfully.
```

Risultato della compilazione.

chiamata alla routine mediante l'istruzione call. A questo punto dobbiamo configurare il modulo LCD e per fare questo dobbiamo inviare un comando al display. Come avviene con l'inizializzazione, esiste già una routine che fa questo, la routine "LCD_REG". Prima di chiamare questa routine è necessario caricare il registro di lavoro W con il valore che vogliamo dare al comando, e di seguito tramite la routine, verrà inviato al display. Configureremo il display in modo da attivare LCD ON, il cursore e il lampeggio. A questo punto ci rimane solamente da inviare il messaggio che vogliamo visualizzare. È necessario inviare carattere per carattere, quindi invieremo prima il carattere "C", dopo "i" e così via fino a completare la parola "Ciao". Per inviare un carattere lo dobbiamo caricare sul registro di lavoro e poi chiamare la routine "LCD_DATO".

Fatto questo il programma deve entrare in un ciclo di attesa in modo che il messaggio rimanga visualizzato sul display.

Compilazione

Nel secondo CD allegato all'opera potrete trovare il codice che abbiamo sviluppato con il nome di "Ciao.asm", all'interno della cartella "LCD". Facciamo partire MPLAB per compilare il programma, creiamo un nuovo progetto che chiameremo "Ciao.pjt" e nella finestra di edit allegheremo il nostro codice al progetto.

Fatto questo apriamo il nostro programma per visualizzarlo sul display e selezioniamo l'opzione Build All per eseguire la compilazione e l'assemblaggio. Come potete vedere nelle immagini il codice si compila senza errori.

Simulazione

Al momento di simulare ricordate che è necessario copiare nella cartella dei progetti tutti i file associati ai programmi. Per simulare correttamente il nostro programma dobbiamo copiare la libreria del modulo LCD nella cartella dei progetti.

Watch_1

Address	Symbol	Value
06	PORTB	B'00111000'
200	W	B'00001010'
05	PORTA	B'00000000'

Finestra per visualizzare le porte e W.

Forziamo i valori delle variabili. aprendo la finestra Modify.

Il simulatore entra in un ciclo all'interno della subroutine di ritardo.

Apriamo le finestre tipiche della simulazione.

Nella visualizzazione dei registri più importanti inseriamo le porte B e A in modo binario e il registro di lavoro W. Ricordate che è necessario scegliere "w" in caratteri minuscoli per visualizzarlo correttamente.

Inizieremo la simulazione e quando arriveremo alla chiamata della subroutine "LCD_INI" il simulatore esegue la routine della libreria. Il simulatore salta alla subroutine avvicinandosi molto a un funzionamento quasi reale del programma, realizzando una simulazione completa dell'applicazione.

Se continuiamo premendo F7, simulando passo a passo verrà eseguita la subroutine, e potremo vedere il simulatore che passerà alle altre subroutine all'interno della libreria, in base allo sviluppo del programma. Saltiamo a "LCD_REG", da questa a "LCD_BUSY", a "LCD_E" e infine a "LCD_DELAY". In quest'ultima subroutine il simulatore si ferma in un ciclo attendendo che la variabile del temporizzatore "Lcd_Temp_2" raggiunga lo zero.

Cambiate la visualizzazione dell'uscita in ASCII.

Dobbiamo forzare il valore di questa variabile e per fare questo apriremo la finestra Modify e forzeremo la variabile a 1 in decimale, come possiamo vedere nella figura. Dobbiamo fare lo stesso con la variabile "Lcd_Temp_1" per poter uscire da questa subroutine. Fatto questo il programma continua a eseguire la routine "LCD_INI", le cui istruzioni si ripetono, dobbiamo quindi mantenere aperta la finestra Modify per poter uscire dalla routine di temporizzazione.

Quando termineremo di eseguire la routine "LCD_INI" la simulazione ritorna al programma principale e continua dalla linea successiva a quella della chiamata alla subroutine. Si configura il display LCD e si inizia a inviare i caratteri al display. Per vedere se i caratteri sono inviati correttamente cambiate il modo di visualizzazione della porta B di uscita. Cliccando con il pulsante sinistro del mouse sull'angolo superiore sinistro della finestra ed editandola, sarà possibile cambiare le proprietà di visualizzazione del registro che abbiamo selezionato. Continuate l'esecuzione della simulazione e verificate che il programma risponda in modo soddisfacente.

Lavorare con un display LCD è facile se si hanno ben chiari i concetti. Questo programma è un chiaro esempio di come sia semplice visualizzare un messaggio sul display. Nei prossimi fascicoli complicheremo un po' gli esercizi per poter ottenere il massimo rendimento dal modulo LCD.